

Entrada/salida de propósito general

Sergio Barrachina Mir Germán Fabregat Llueca

Departamento de Ingeniería y Ciencia de los Computadores
Universidad Jaume I

Contenido

- 1 Entrada/salida de propósito general (GPIO)
- 2 Gestión del tiempo

Características de la GPIO

Comunica el procesador con el exterior.

Permite:

- Configurar pines como entrada.
- Configurar pines como salida.
- Leer pines de entrada.
- Escribir en los pines de salida.



Microcontrolador ATSAM3X8E
ARM Cortex-M3
144 pines (103 pines GPIO)

Características de la GPIO

Comunica el procesador con el exterior.

Permite:

- Configurar pines como entrada.
- Configurar pines como salida.
- Leer pines de entrada.
- Escribir en los pines de salida.



Microcontrolador ATSAM3X8E
ARM Cortex-M3
144 pines (103 pines GPIO)



¿Para qué se utiliza?

En computadores

- Para leer pulsadores (Power, apertura chasis).
- Para encender/apagar los LED del chasis.

En sistemas empotrados

- Para gestionar la entrada/salida.

¿Para qué se utiliza?

En computadores

- Para leer pulsadores (Power, apertura chasis).
- Para encender/apagar los LED del chasis.

En sistemas empotrados

- Para gestionar la entrada/salida.

¿Cómo se utiliza?

Programa que consulta el estado de un pulsador (en el bit 20, del puerto B) y enciende un LED (bit 12) si estaba pulsado:

```
1      ldr    r7, =PB           @ Dirección del puerto
2      ldr    r6, =0x00100000 @ Máscara para el bit 20
3      ldr    r0, [r7]         @ Leemos el puerto
4      ands  r0, r6            @ y verificamos el bit
5      beq   sigue            @ Seguimos si está a 0
6      ldr    r6, =0x00001000 @ Máscara para el bit 12
7      ldr    r0, [r7]         @ Leemos el puerto
8      orr   r0, r6            @ ponemos a 1 el bit
9      str   r0, [r7]         @ y lo escribimos en el puerto
```

¿Cómo se utiliza?

Programa que consulta el estado de un pulsador (en el bit 20, del puerto B) y enciende un LED (bit 12) si estaba pulsado:

```
1      ldr    r7, =PB           @ Dirección del puerto
2      ldr    r6, =0x00100000 @ Máscara para el bit 20
3      ldr    r0, [r7]         @ Leemos el puerto
4      ands  r0, r6           @ y verificamos el bit
5      beq   sigue           @ Seguimos si está a 0
6      ldr    r6, =0x00001000 @ Máscara para el bit 12
7      ldr    r0, [r7]         @ Leemos el puerto
8      orr   r0, r6           @ ponemos a 1 el bit
9      str   r0, [r7]         @ y lo escribimos en el puerto
```



¿Pines de entrada o salida?

Previamente al código anterior será necesario especificar por medio de un **registro de control** asociado al puerto B que:

- El pin 20 es de entrada y
- el pin 12 es de salida.

Si el pin es de entrada. . .

Su valor depende del circuito conectado a dicho pin.

Si el pin es de salida. . .

Su valor depende del programa.

¿Pines de entrada o salida?

Previamente al código anterior será necesario especificar por medio de un **registro de control** asociado al puerto B que:

- El pin 20 es de entrada y
- el pin 12 es de salida.

Si el pin es de entrada. . .

Su valor depende del circuito conectado a dicho pin.

Si el pin es de salida. . .

Su valor depende del programa.

¿Pines de entrada o salida?

Previamente al código anterior será necesario especificar por medio de un **registro de control** asociado al puerto B que:

- El pin 20 es de entrada y
- el pin 12 es de salida.

Si el pin es de entrada. . .

Su valor depende del circuito conectado a dicho pin.

Si el pin es de salida. . .

Su valor depende del programa.



¿Registros asociados a un puerto?

Registro de control

Para indicar la dirección de los pines.

Registro de datos de salida

Para escribir en los pines de salida.

Registro de datos de entrada

Para leer el valor presente en los pines de entrada.

¿Registros asociados a un puerto?

Registro de control

Para indicar la dirección de los pines.

Registro de datos de salida

Para escribir en los pines de salida.

Registro de datos de entrada

Para leer el valor presente en los pines de entrada.

¿Registros asociados a un puerto?

Registro de control

Para indicar la dirección de los pines.

Registro de datos de salida

Para escribir en los pines de salida.

Registro de datos de entrada

Para leer el valor presente en los pines de entrada.

Leer de la salida / Escribir en la entrada

¿Qué ocurre si leemos de un pin de salida o escribimos en uno de salida?

Depende del diseño electrónico de los circuitos de E/S del microcontrolador.

Leer de la salida: se lee el valor en el registro de salida o el valor en la propia salida (que deberían coincidir).

Escribir en la entrada: se suele escribir en el registro de salida, pero no se modifica el valor del pin (sirve para inicializar el valor de salida antes de activar el pin como salida).

Leer de la salida / Escribir en la entrada

¿Qué ocurre si leemos de un pin de salida o escribimos en uno de salida?

Depende del diseño electrónico de los circuitos de E/S del microcontrolador.

Leer de la salida: se lee el valor en el registro de salida o el valor en la propia salida (que deberían coincidir).

Escribir en la entrada: se suele escribir en el registro de salida, pero no se modifica el valor del pin (sirve para inicializar el valor de salida antes de activar el pin como salida).

Leer de la salida / Escribir en la entrada

¿Qué ocurre si leemos de un pin de salida o escribimos en uno de salida?

Depende del diseño electrónico de los circuitos de E/S del microcontrolador.

Leer de la salida: se lee el valor en el registro de salida o el valor en la propia salida (que deberían coincidir).

Escribir en la entrada: se suele escribir en el registro de salida, pero no se modifica el valor del pin (sirve para inicializar el valor de salida antes de activar el pin como salida).

Leer de la salida / Escribir en la entrada

¿Qué ocurre si leemos de un pin de salida o escribimos en uno de salida?

Depende del diseño electrónico de los circuitos de E/S del microcontrolador.

Leer de la salida: se lee el valor en el registro de salida o el valor en la propia salida (que deberían coincidir).

Escribir en la entrada: se suele escribir en el registro de salida, pero no se modifica el valor del pin (sirve para inicializar el valor de salida antes de activar el pin como salida).

Relación entre magnitudes físicas y niveles lógicos

Por si estáis interesados, en el apartado se describe con detalle la relación entre magnitudes físicas y los niveles lógicos 0 y 1.

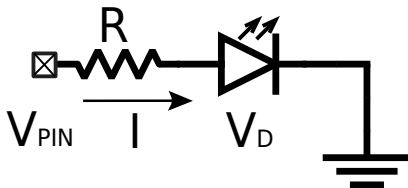
¡No forma parte de los objetivos de la asignatura!

Relación entre magnitudes físicas y niveles lógicos

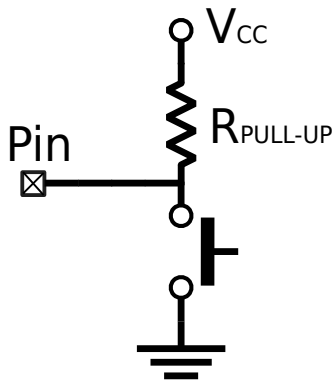
Por si estáis interesados, en el apartado se describe con detalle la relación entre magnitudes físicas y los niveles lógicos 0 y 1.

¡No forma parte de los objetivos de la asignatura!

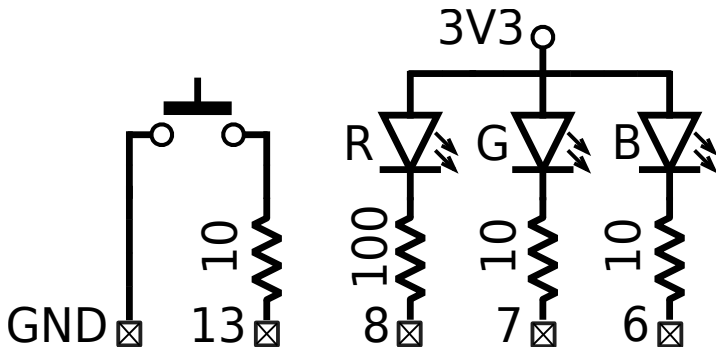
Conexión de un LED a un pin de salida



Conexión de un pulsador a un pin de entrada

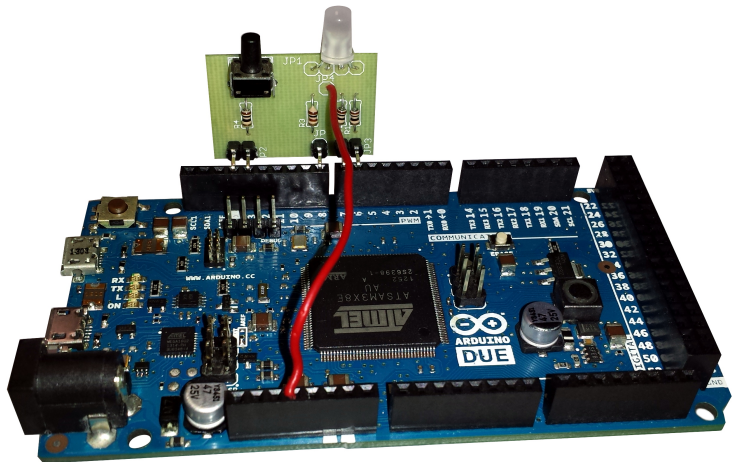


Configuración utilizada en el laboratorio



El pin 13 se configura como entrada con *pull-up*.
Los pines 6, 7 y 8 como salida (los LED en ánodo común).

Tarjeta de E/S conectada al Arduino DUE



Contenido

- 1 Entrada/salida de propósito general (GPIO)
- 2 Gestión del tiempo

Gestión del tiempo: ¿Para qué?

- Fecha y hora de modificación de un archivo.
- Poner en marcha tareas con cierta periodicidad.
- Determinar si una tecla se ha pulsado durante más de x tiempo.
- Calcular el tiempo transcurrido entre dos eventos.

Dispositivos habituales para la gestión del tiempo

- Temporizador (*timer*).
- Reloj de tiempo real (*real time clock, RTC*).

Temporizador (1)

- El dispositivo más común y sencillo.
- Es un registro contador.
- Se incrementa de forma periódica a partir de una señal de reloj.
- La resolución viene dada por el periodo de actualización.
- El tiempo máximo viene dado por:
 - Número de bits del registro.
 - Periodo de actualización.
- El periodo de actualización se puede variar por medio de un *divisor de frecuencia*.



Temporizador (1)

- El dispositivo más común y sencillo.
- Es un registro contador.
- Se incrementa de forma periódica a partir de una señal de reloj.
- La resolución viene dada por el periodo de actualización.
- El tiempo máximo viene dado por:
 - Número de bits del registro.
 - Periodo de actualización.
- El periodo de actualización se puede variar por medio de un *divisor de frecuencia*.



Temporizador (1)

- El dispositivo más común y sencillo.
- Es un registro contador.
- Se incrementa de forma periódica a partir de una señal de reloj.
- La resolución viene dada por el periodo de actualización.
- El tiempo máximo viene dado por:
 - Número de bits del registro.
 - Periodo de actualización.
- El periodo de actualización se puede variar por medio de un *divisor de frecuencia*.

Temporizador (1)

- El dispositivo más común y sencillo.
- Es un registro contador.
- Se incrementa de forma periódica a partir de una señal de reloj.
- La resolución viene dada por el periodo de actualización.
- El tiempo máximo viene dado por:
 - Número de bits del registro.
 - Periodo de actualización.
- El periodo de actualización se puede variar por medio de un *divisor de frecuencia*.

Temporizador (1)

- El dispositivo más común y sencillo.
- Es un registro contador.
- Se incrementa de forma periódica a partir de una señal de reloj.
- La resolución viene dada por el periodo de actualización.
- El tiempo máximo viene dado por:
 - Número de bits del registro.
 - Periodo de actualización.
- El periodo de actualización se puede variar por medio de un *divisor de frecuencia*.

Temporizador (1)

- El dispositivo más común y sencillo.
- Es un registro contador.
- Se incrementa de forma periódica a partir de una señal de reloj.
- La resolución viene dada por el periodo de actualización.
- El tiempo máximo viene dado por:
 - Número de bits del registro.
 - Periodo de actualización.
- El periodo de actualización se puede variar por medio de un *divisor de frecuencia*.

Temporizador (II)

- El temporizador se suele utilizar para medir el tiempo transcurrido entre dos eventos.
- P.e., entre el inicio y el fin de un programa:
$$Tiempo = (Temporizador_{fin} - Temporizador_{inicio}) \cdot Periodo$$
- Si el temporizador se desborda → ERROR:
Desbordamiento del contador → interrupción.

Temporizador (II)

- El temporizador se suele utilizar para medir el tiempo transcurrido entre dos eventos.
- P.e., entre el inicio y el fin de un programa:
$$\text{Tiempo} = (\text{Temporizador}_{fin} - \text{Temporizador}_{inicio}) \cdot \text{Periodo}$$
- Si el temporizador se desborda → ERROR:
Desbordamiento del contador → interrupción.

Temporizador (II)

- El temporizador se suele utilizar para medir el tiempo transcurrido entre dos eventos.
- P.e., entre el inicio y el fin de un programa:
$$Tiempo = (Temporizador_{fin} - Temporizador_{inicio}) \cdot Periodo$$
- Si el temporizador se desborda \rightarrow ERROR:
Desbordamiento del contador \rightarrow interrupción.

Reloj de tiempo real

- Dispositivo encargado de mantener la fecha y hora actuales, incluso cuando el computador está desconectado de la alimentación.
- Emplea como frecuencia base 32.768 Hz (2^{15})
→ el bit 15 cambia de valor cada segundo.
- Suele ser un dispositivo independiente formado por:
 - Circuito oscilador,
 - registro contador,
 - pequeña cantidad de memoria RAM,
 - batería (que alimenta a todo lo anterior).



Reloj de tiempo real

- Dispositivo encargado de mantener la fecha y hora actuales, incluso cuando el computador está desconectado de la alimentación.
- Emplea como frecuencia base 32.768 Hz (2^{15})
→ el bit 15 cambia de valor cada segundo.
- Suele ser un dispositivo independiente formado por:
 - Circuito oscilador,
 - registro contador,
 - pequeña cantidad de memoria RAM,
 - batería (que alimenta a todo lo anterior).



Reloj de tiempo real

- Dispositivo encargado de mantener la fecha y hora actuales, incluso cuando el computador está desconectado de la alimentación.
- Emplea como frecuencia base 32.768 Hz (2^{15})
→ el bit 15 cambia de valor cada segundo.
- Suele ser un dispositivo independiente formado por:
 - Circuito oscilador,
 - registro contador,
 - pequeña cantidad de memoria RAM,
 - batería (que alimenta a todo lo anterior).



Reloj de tiempo real

- Dispositivo encargado de mantener la fecha y hora actuales, incluso cuando el computador está desconectado de la alimentación.
- Emplea como frecuencia base 32.768 Hz (2^{15})
→ el bit 15 cambia de valor cada segundo.
- Suele ser un dispositivo independiente formado por:
 - Circuito oscilador,
 - registro contador,
 - pequeña cantidad de memoria RAM,
 - batería (que alimenta a todo lo anterior).



Reloj de tiempo real

- Dispositivo encargado de mantener la fecha y hora actuales, incluso cuando el computador está desconectado de la alimentación.
- Emplea como frecuencia base 32.768 Hz (2^{15})
→ el bit 15 cambia de valor cada segundo.
- Suele ser un dispositivo independiente formado por:
 - Circuito oscilador,
 - registro contador,
 - pequeña cantidad de memoria RAM,
 - batería (que alimenta a todo lo anterior).



Entrada/salida de propósito general

Sergio Barrachina Mir Germán Fabregat Llueca

Departamento de Ingeniería y Ciencia de los Computadores
Universidad Jaume I